



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/754,890	01/05/2001	K. Rustan M. Leino	9772-0274-999	4747

7590 11/20/2003

PENNIE & EDMONDS LLP  
3300 Hillview Avenue  
Palo Alto, CA 94304

EXAMINER

YIGDALL, MICHAEL J

ART UNIT	PAPER NUMBER
----------	--------------

2122

DATE MAILED: 11/20/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

**Office Action Summary**

Application No.

09/754,890

Applicant(s)

LEINO ET AL.

Examiner

Michael J. Yigdall

Art Unit

2122

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 05 January 2001.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-46 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-46 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 05 January 2001 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- 11) ☐ The proposed drawing correction filed on \_\_\_\_\_ is: a) ☐ approved b) ☐ disapproved by the Examiner.
- If approved, corrected drawings are required in reply to this Office action.
- 12) ☐ The oath or declaration is objected to by the Examiner.

**Priority under 35 U.S.C. §§ 119 and 120**

- 13) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.
- 14) ☒ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).
- a) ☐ The translation of the foreign language provisional application has been received.
- 15) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

**Attachment(s)**

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) 2,5.
- 4) ☐ Interview Summary (PTO-413) Paper No(s). \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other:

### **DETAILED ACTION**

1. Claims 1-46 are pending and have been examined. The priority date considered for the application is 7 January 2000.

#### ***Claim Rejections - 35 USC § 112***

2. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

3. Claims 34 and 35 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 34 recites the limitation "the flow control label name" in line 1. There is insufficient antecedent basis for this limitation in the claim because claim 23 recites a plurality of control labels. The intended parent claim may be claim 33, rather than claim 23 as recited.

Claim 35 recites the limitation "the flow control label name" in line 1. There is insufficient antecedent basis for this limitation in the claim because claim 23 recites a plurality of control labels. The intended parent claim may be claim 33, rather than claim 23 as recited.

#### ***Claim Rejections - 35 USC § 103***

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

5. Claims 1-5, 8, 11, 14, 17, 20, 22-27, 30, 33, 36, 39, 42 and 45 are rejected under 35 U.S.C. 103(a) as being unpatentable over "Extended Static Checking" by Detlefs et al. (hereinafter Detlefs) in view of U.S. Pat. No. 4,920,538 to Chan et al. (hereinafter Chan).

With respect to claim 1, Detlefs discloses a method of verifying whether a specified computer program satisfies a predefined set of conditions (see page 24, Fig. 10), comprising:

(a) converting the program into a logical equation representing the predefined set of conditions as applied to instructions and elements of the instructions of the program (see page 23, section 4, paragraph 1, lines 1-4, which shows generating a logical equation based on the conditions annotated in the program).

Detlefs does not expressly disclose the limitation wherein the converting step includes inserting flow control labels into the sub-equations of the logical equation, the flow control labels identifying conditional branch points in the specified computer program.

Detlefs does show that any sub-equation of the logical equation can be given a label (see page 29, section 6, paragraph 2, lines 1-2).

Chan discloses the limitation above (see column 1, lines 38-53, which shows inserting control markers into the code to identify conditional branch points) for the purpose of checking execution paths and verifying conditional branches (see column 1, lines 26-35).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Detlefs with the labeling feature taught by Chan, for the purpose of verifying conditional branches.

Detlefs further discloses:

(b) applying a theorem prover to the logical equation to determine the truth of the logical equation, and when the truth of the logical equation cannot be proved, generating at least one counter-example identifying one of the conditions, one or more variable values inconsistent with the one condition, and any of the flow control labels for conditional branch points of the program associated with the identified variable values (see page 23, section 4, paragraph 1, lines 4-8, which shows using a theorem prover to determine when the truth of the logical equation does not hold; see also page 29, section 6, paragraph 2, lines 3-7, which shows generating a counter-example comprising the labels that identify inconsistent conditions in the equation); and

(c) converting the at least one counter-example into an error message that includes a program trace when the counter-example identifies one or more of the flow control labels (see page 29, section 6, paragraph 2, lines 3-4 and 7-9, which shows translating a counter-example into an error message that traces the source of the error using the flow control labels).

With respect to claim 2, Detlefs further discloses the limitation wherein the converting step additionally comprises a step of converting the program into an intermediate language form prior to creating the logical equation (see page 29, section 6, paragraph 1, lines 7-9, which shows converting the program into Dijkstra's guarded commands, i.e. an intermediate language, before generating the logical equation).

With respect to claim 3, Detlefs further discloses the limitation wherein the flow control labels are inserted before converting the program into the intermediate language form (see page 29, section 6, paragraph 1, lines 7-9, which shows converting an annotated program into an intermediate language; note that the annotations could comprise labels that are inserted before converting the program).

With respect to claim 4, Detlefs further discloses the limitation wherein the flow control labels are inserted after converting the program into the intermediate language form (see page 29, section 6, paragraph 2, lines 6-7, which shows that the verification condition generator inserts labels into the logical equation; note that because the program is converted into the intermediate language prior to generating the logical equation, the labels would be inserted after converting the program).

With respect to claim 5, Detlefs further discloses the limitation wherein the intermediate language form is Dijkstra's guarded command language (see page 29, section 6, paragraph 1, lines 7-9, which shows using Dijkstra's guarded commands).

With respect to claim 8, Detlefs does not expressly disclose the limitation wherein at least one of the flow control labels is in the form  $L \implies P$  wherein L is a flow control label name and P is a subcomponent of the logical equation.

Detlefs does show using Dijkstra's guarded commands and weakest-precondition equations to generate a verification condition (see page 29, section 6, paragraph 1, lines 7-9). Detlefs also shows using flow control labels in the verification condition, and using a theorem prover to determine the truth of the logical equation and to find a counter-example (see page 29, section 6, paragraph 2, lines 1-9).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to implement the system of Detlefs using a mathematical notation such as  $L \implies P$ . The choice of notation used in the label names does not depart from the teachings of Detlefs, because the functionality remains substantially the same.

With respect to claim 11, Detlefs does not expressly disclose the limitation wherein at least one of the flow control labels is in the form  $L=k \implies P$  wherein L is a flow control label name, k is a constant value and P is a subcomponent of the logical equation.

Detlefs does show using Dijkstra's guarded commands and weakest-precondition equations to generate a verification condition (see page 29, section 6, paragraph 1, lines 7-9). Detlefs also shows using flow control labels in the verification condition, and using a theorem prover to determine the truth of the logical equation and to find a counter-example (see page 29, section 6, paragraph 2, lines 1-9).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to implement the system of Detlefs using a mathematical notation such as  $L=k \implies P$ . The choice of notation used in the label names does not depart from the teachings of Detlefs, because the functionality remains substantially the same.

With respect to claim 14, Detlefs does not expressly disclose the limitation wherein at least one of the flow control labels is in the form {LBLPOS L P} wherein L is a flow control label name and P is a subcomponent of the logical equation.

Detlefs does show using Dijkstra's guarded commands and weakest-precondition equations to generate a verification condition (see page 29, section 6, paragraph 1, lines 7-9). Detlefs also shows using flow control labels in the verification condition, and using a theorem prover to determine the truth of the logical equation and to find a counter-example (see page 29, section 6, paragraph 2, lines 1-9).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to implement the system of Detlefs using a mathematical notation such as {LBLPOS L

P}. The choice of notation used in the label names does not depart from the teachings of Detlefs, because the functionality remains substantially the same.

With respect to claim 17, Detlefs does not expressly disclose the limitation wherein at least one of the flow control labels is in the form  $\neg\{\text{LBLNEG } L \neg P\}$  wherein L is a flow control label name and P is a subcomponent of the logical equation.

Detlefs does show using Dijkstra's guarded commands and weakest-precondition equations to generate a verification condition (see page 29, section 6, paragraph 1, lines 7-9). Detlefs also shows using flow control labels in the verification condition, and using a theorem prover to determine the truth of the logical equation and to find a counter-example (see page 29, section 6, paragraph 2, lines 1-9).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to implement the system of Detlefs using a mathematical notation such as  $\neg\{\text{LBLNEG } L \neg P\}$ . The choice of notation used in the label names does not depart from the teachings of Detlefs, because the functionality remains substantially the same.

With respect to claim 20, Detlefs does not expressly disclose the limitation wherein at least one of the flow control labels is in the form  $\{\text{LBLPOS } L \text{ True}\} \implies P$  wherein L is a flow control label name and P is a subcomponent of the logical equation.

Detlefs does show using Dijkstra's guarded commands and weakest-precondition equations to generate a verification condition (see page 29, section 6, paragraph 1, lines 7-9). Detlefs also shows using flow control labels in the verification condition, and using a theorem prover to determine the truth of the logical equation and to find a counter-example (see page 29, section 6, paragraph 2, lines 1-9).



It would have been obvious to one of ordinary skill in the art at the time the invention was made to implement the system of Detlefs using a mathematical notation such as  $\{\text{LBLPOS L True}\} \Rightarrow P$ . The choice of notation used in the label names does not depart from the teachings of Detlefs, because the functionality remains substantially the same.

With respect to claim 22, Detlefs further discloses the limitation wherein the flow control label name identifies a line number in the specified computer program at which an associated program instruction is located (see page 29, section 6, paragraph 2, lines 7-8, which shows that the name of the label identifies a source position, i.e. a line number in the program).

Detlefs does not expressly disclose the limitation wherein the flow control label includes a sequence number indicating an order of execution of the program instruction at the identified line number relative to other program instructions identified by other flow control labels.

Chan further discloses the limitation above (see column 1, lines 38-53, which shows using control markers to identify branches in a program; see also column 1, lines 54-57, which shows sequence identifiers used for each execution path) for the purpose of checking execution paths and verifying conditional branches (see column 1, lines 26-35).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Detlefs with the labeling feature taught by Chan, for the purpose of verifying conditional branches.

With respect to claim 23, see the explanation for claim 1 above. Note that Detlefs further discloses that the system is implemented as a computer program product (see page 23, section 4, paragraph 2, lines 1-5). Also note that Detlefs shows a verification condition generator, a

theorem prover, control labels, and a post processor (see page 24, Fig. 10, and page 29, section 6, paragraph 2, lines 1-4).

With respect to claim 24, see the explanation for claim 2 above.

With respect to claim 25, see the explanation for claim 3 above.

With respect to claim 26, see the explanation for claim 4 above.

With respect to claim 27, see the explanation for claim 5 above.

With respect to claim 30, see the explanation for claim 8 above.

With respect to claim 33, see the explanation for claim 11 above.

With respect to claim 36, see the explanation for claim 14 above.

With respect to claim 39, see the explanation for claim 17 above.

With respect to claim 42, see the explanation for claim 20 above.

With respect to claim 45, see the explanation for claim 22 above.

6. Claims 6, 7, 9, 10, 12, 13, 15, 16, 18, 19, 21, 28, 29, 31, 32, 34, 35, 37, 38, 40, 41, 43, 44 and 46 are rejected under 35 U.S.C. 103(a) as being unpatentable over Detlefs in view of Chan as applied to claims 1-5, 8, 11, 14, 17, 20, 22-27, 30, 33, 36, 39, 42 and 45 above, and further in view of U.S. Pat. No. 5,854,924 to Rickel et al. (hereinafter Rickel).

With respect to claim 6, Detlefs further discloses the limitation wherein at least one of the flow control labels includes a flow control label name that includes a string that identifies a line number in the specified computer program (see page 29, section 6, paragraph 2, lines 7-8, which shows that the name of the label includes a source position, i.e. a line number in the program; note that a label name is inherently a string).

Detlefs does not expressly disclose the limitation wherein the string identifies a type of branch in the program. However, Detlefs does show that the string identifies a type of error (see page 29, section 6, paragraph 2, lines 7-8).

Rickel discloses the limitation above (see column 8, lines 20-23, which shows using branch labels in the intermediate language form of a program, and lines 47-53, which shows that the intermediate file notes every flow path in the program, based on a branch type such as a function call or a jump table), for the purpose of enabling a static debugger to traverse all the paths in a program to find errors (see column 8, lines 61-67, and column 9, lines 1-3).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Detlefs with the labeling feature taught by Rickel, for the purpose of statically checking a program for errors.

With respect to claim 7, Detlefs does not expressly disclose the limitation wherein at least one of the flow control labels includes a flow control label name that includes a value associated with an entry in a table that identifies a type of branch in the program and a line number in the specified computer program.

Detlefs does show a flow control label name that identifies a line number in the specified computer program (see page 29, section 6, paragraph 2, lines 7-8, which shows that the name of the label identifies a source position, i.e. a line number in the program).

Rickel further discloses the limitation above (see Fig. 7, which shows using jump tables, and jump targets from the symbol table, as branch labels), for the purpose of enabling a static debugger to traverse all the paths in a program to find errors (see column 8, lines 61-67, and column 9, lines 1-3).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Detlefs with the labeling feature taught by Rickel, for the purpose of statically checking a program for errors.

With respect to claims 9, 12, 15, 18 and 21, see the explanation for claim 6 above.

With respect to claims 10, 13, 16, 19 and 46, see the explanation for claim 7 above.

With respect to claim 28, 31, 34, 37, 40 and 43, see the explanation for claim 6 above.

With respect to claim 29, 32, 35, 38, 41 and 44, see the explanation for claim 7 above.

### ***Conclusion***

7. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. U.S. Pat. No. 5,695,539 to Haley et al. discloses a method for detecting errors in computer programs. U.S. Pat. No. 5,909,580 to Crelier et al. discloses a method for reporting the locations of problems in computer programs. U.S. Pat. No. 6,029,022 to Afifi et al. discloses a code analyzer comprising an intermediate language and weakest precondition. U.S. Pat. No. 6,128,774 to Necula et al. discloses a system for verifying software by solving proofs. U.S. Pat. No. 6,374,368 to Mitchell et al. discloses weakest precondition analysis. U.S. Pat. No. 6,560,774 to Gordon et al. discloses verifying an intermediate language.

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Yigdall whose telephone number is (703) 305-0352. The examiner can normally be reached on Monday through Friday from 8:00am to 4:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (703) 305-4552. The fax phone number for the organization where this application or proceeding is assigned is (703) 746-7239.

Application/Control Number: 09/754,890  
Art Unit: 2122

Page 12

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

MY

Michael J. Yigdall  
Examiner  
Art Unit 2122

mjy  
November 6, 2003



JOHN CHAVIS  
PATENT EXAMINER  
ART UNIT 2124